TITLE OF THE INVENTION

METHOD TO IMPROVE PERFORMANCE AND REDUCE COMPLEXITY OF TURBO DECODER

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit U.S. Provisional Application Serial Number 60/250,201, filed November 30, 2000.

BACKGROUND OF THE INVENTION

[0002] The invention relates to iterative decoding methods and iterative decoders. In particular, the invention relates to iterative decoding methods and iterative decoders used in conjunction with communication systems.

[0003] Iterative decoding schemes are well known in the art. Typically, these decoding schemes are used in receivers to decode encoded data that is transmitted across a channel from a transmitter to a receiver. One type of iterative decoding scheme is known as turbo decoding, which performs several iterations on each frame of received data until the data is properly decoded.

[0004] Several systems incorporate turbo decoders, such as, for example 3rd generation wireless communication systems and satellite communication systems.

[0005] For example, the channel coding scheme used in 3rd generation wireless systems is the PCCC (Parallel Concatenated Convolution Code) turbo code. The decoding algorithm is an iterative decoding algorithm. The PCCC turbo encoder structure as specified by 3GPP (3rd Generation Partnership Project) for the W-CDMA 3rd generation wireless systems is as shown below.

[0006] The transfer function of the 8-state constituent code for PCCC is

$$G(D)=\left[1,\frac{n(D)}{d(D)}\right]$$

where,

$$d(D)=1+D^2+D^3$$

$$n(D)=1+D+D^3.$$

[0007] The initial value of the PCCC encoder shift registers are zeros. The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate,

1/3. For a 1/3 code rate, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc. Trellis termination is performed by taking the tail bits from the shift register feedback after all the information bits are encoded. Tail bits are added after the encoding of information bits. The first three tail bits shall be used to terminate the first constituent encoder (upper switch of Figure 1 in a lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of Figure 1 in a lower position) while the first constituent encoder is disabled. The transmitted bits for trellis termination shall then be

X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2).

[0008] The Turbo code internal interleaver can be the PIL interleaver. The two most common decoding algorithms are the maximum a posteriori (MAP) algorithm and the soft output Viterbi algorithm (SOVA). The MAP algorithm gives the best performance but the SOVA algorithm provides reduced complexity. There are two forms of MAP algorithms in use for decoding the PCCC. These are the Bahl, Cocke, Jelenik and Raviv (BCJR) algorithm and the soft input soft output (SISO) algorithm. The SISO algorithm is essentially the same as the BCJR algorithm but has a more elegant form and has greater flexibility. A block diagram of the PCCC decoder is shown in Figure 2.

[0009] The basic unit of the algorithm is the SISO module. The SISO module is a four port device that has two inputs and two outputs and which module implements the SISO algorithm. The module's inputs are the probabilities of the information symbols P(u; I) and code symbols P(c; I) labeling the edges of the code trellis, and the outputs are updates of these probabilities based upon knowledge of the trellis, P(u; O) and P(c; O), respectively.

[0010] Figure 2 shows the use of the SISO module in a PCCC decoder. The inputs to the first SISO module are the probabilities of the code symbols and information symbols for the first code. The code symbol probabilities are similar to the branch metrics used in a Viterbi decoder and are determined from the demodulated signal amplitude and an estimate of the noise variance. The information symbol probabilities for the first code are the de-interleaved updated information symbol probabilities of the second code from the previous iteration (during the first iteration nothing is known about these probabilities, so the a priori distribution should be used).

[0011] Only the updated probabilities of the information symbols of the first code are used while the updated code symbol probabilities are ignored. The updated probabilities of the first code's information symbols are interleaved and become the input to the second SISO module, along with the second code's code symbol probabilities, again determined from the output of the demodulator and an estimate of the noise variance (note that only the parity bits from the second encoder are transmitted, so the probability of the punctured systematic bits are all 0:5). Again, the code symbol output is not used, and the probabilities of the information symbols from the second code are de-interleaved to become the input to the first SISO module during the next iteration.

[0012] After all iterations are complete, the probabilities of the information symbols from both codes can be combined by multiplying the probabilities. A final decision can then be made by choosing the information symbols with the highest probabilities.

[0013] The known decoders have several drawbacks. For instance, the performance of turbo decoders improve considerably as the number of iterations increase but the time taken for decoding also increases as the number of iterations increase. In particular, the time constraints for decoding (decoding time constraint per frame) in the W-CDMA system necessitates that the number of iterations that the system can perform within the given time constraints be optimized to the fullest. Typically, the system's decoding time constraint per frame is limited by the system's transmission and reception rate of frames so that the decoding time per frame does not exceed the transmission time per frame.

[0014] Although some decoders have addressed these concerns by fixing the number of iterations, such a solution can have drawbacks. A slow decoder may not be able to support the number of iterations required to achieve a particular performance level. A faster decoder, on the other hand, may be able to support a particular performance level, but the faster decoder will also perform a number of redundant iterations because each frame may not require the same number of iterations to achieve the same performance.

[0015] For example, if a performance level requires 12 iterations to be performed for a frame length of 1280, a decoder which can only support 3 iterations within the time constraint for that frame length will severely degrade performance.

SUMMARY OF INVENTION

[0016] Pursuant to a first aspect of the invention, in order to make the number of iterations per frame variable, the use of an error detection check along with an iterative decoder may be used. In one method and embodiment, the error detection check can be performed after every iteration, in others, it may be repeatedly performed after a predetermined number of iterations. The error detection check enables a user to determine how many iterations are required for incoming frames and, consequently, allows the user to determine when to stop the iterative decoding process at the receiver. One error detection check that may be employed is the Cyclic Redundancy Check (CRC).

[0017] Pursuant to the first aspect of the invention, incoming frames can be stored (or buffered) when the decoder is busy processing an earlier frame. Thus, methods and embodiments of the first aspect of the invention allow for decoding of frames that require decoding for a period longer than the system's decoding time constraint per frame. At least one other frame will be subsequently decoded in a period shorter than the decoding time constraint. Therefore, pursuant to methods and embodiments of the invention, the average decoding time period per frame will be less than or equal to the system's decoding time constraint per frame.

[0018] Pursuant to a second aspect of the invention, an optimal storage device size or buffer length to be used in embodiments and methods incorporating the first aspect of the invention is determined. The optimal storage device size or buffer length reduces buffer or storage device frame overflow.

[0019] Pursuant to a third aspect of the invention, an alternate storage device or alternate buffer can be used to store frames that still have errors after a predetermined number of decoding iterations have been performed on them. Such partially decoded frames are decoded out of sequence at a subsequent time. The frames are then resequenced pursuant to the third aspect of the invention.

[0020] Pursuant to a fourth aspect of the invention, the decoded frames can be retrieved by an upper layer at a constant rate, or pursuant to any process or scheme whose average rate is equal to or less than the rate frames are decoded, even though the decoder decodes the frames at a variable rate. Pursuant to the fourth aspect, sufficiently decoded frames, that, for example, are substantially converged, can be stored in an output storage device or output buffer, and later sent to an upper layer.

BRIEF DESCRIPTION OF THE FIGURES

[0021] Figure 1 depicts a block diagram of a 3G PCCC encoder;

[0022] Figure 2 depicts a block diagram of a PCCC decoder that uses an SISO module;

[0023] Figure 3 depicts an overall system for an AWGN channel, and one configuration of receiver components pursuant to aspects of the invention;

[0024] Figure 4 depicts an alternate configuration of receiver components of FIG. 3 pursuant to aspects of the invention;

[0025] Figure 5 depicts a block diagram of a turbo decoder of FIG. 3 along with a CRC check;

[0026] Figure 6 depicts a BER versus FER plot that may be provided by the system of FIG. 3 for a frame length of 1280 bits after performing 12 iterations for each frame;

[0027] Figure 7 shows the average number of iterations that are required to achieve the performance describe in Figure 6;

[0028] Figure 8 shows the simulation results of the number of iterations taken by the system of FIG. 3 for frames of length 1280 bits and for a 0.5 dB Eb/No value;

[0029] Figure 9 shows the simulation results of the number of iterations taken by the system of FIG. 3 for frames of length 1280 bits and for a 0.75 dB Eb/No value;

[0030] Figure 10 shows the simulation results of the number of iterations taken by the system of FIG. 3 for frames of length 1280 bits and for a 1.0 dB Eb/No value;

[0031] Figure 11 depicts a graph that plots FER as a function of the number of iterations performed by the system of FIG. 3;

[0032] Figure 12 shows a plot of the input buffer occupancy given a turbo decoder of the system of FIG. 3 that is capable of performing 4-4.5 iterations for a frame length of 1280 bits within 10 ms.;

[0033] Figure 13 shows the maximum buffer length required for the same Fb at different processing speeds of the system of FIG. 3;

[0034] Figure 14 depicts a plot for the input waiting time profile of the system of FIG. 3;

[0035] Figure 15 depicts a buffer drop profile that will result if the maximum input buffer size is maintained as the profile shown in Figure 13; and

[0036] Figure 16 depicts a block diagram of yet another configuration of receiver components of the system of FIG. 3 pursuant to aspects of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0037] While the present invention is susceptible of use in various embodiments, it is shown in the drawings and will hereinafter be described several embodiments with the understanding that the present disclosure is to be considered an exemplification of the invention and is not intended to limit the invention to the embodiments illustrated.

[0038] A first aspect of the invention is directed to a method and apparatus for decoding a sequence of frames in a communication system having a predetermined decoding time constraint per frame (e.g. the transmission frame rate of an associated communication system). Pursuant to the first aspect of the invention, a first frame of the sequence of frames is decoded in a time period longer than the predetermined time constraint, and at least one other frame of the sequence of frames is decoded in a time period less than the predetermined time constraint. This allows for an average decoding time of all decoded frames in a period to be less than or equal to the predetermined time constraint of the system.

[0039] Note that all aspects of the invention may be used wherever iterative codes, such as, turbo codes for example, are used. For better error protection, these iterative codes are typically used as channel coding or decoding schemes in communication systems, particularly wireless or satellite communication systems. The invention is particularly useful in situations in which the computational complexity required to obtain optimal performance is beyond system capabilities due to speed constraints of the processor of the decoder. The invention allows the system to obtain the same performance level with a much lower average complexity and a nominal storage or buffer requirement. 3rd Generation wireless systems is but one example of the situations described above. Although several figures and results that show how the invention performs when simulated under AWGN channel conditions are shown, it is contemplated that the invention may be used in conjunction with a variety of different types of channels used in various systems not depicted.

[0040] The overall communication system 20 including an AWGN channel 22 is shown in Figure 3, and Figure 4 shows an alternate configuration of the receiver components shown in Figure 3. For example, Figure 4 shows a decoder 24, error check 26, input storage device 28, and output storage device 30.

[0041] Figure 3 shows an input frame 32 forwarded to a Cyclic Redundancy Check encoder 34 and a turbo encoder 36. The encoded frame is then transmitted across a channel, a AWGN channel 22 in Figure 3, and into a receiver (not shown). Portions of the receiver pertinent to the invention are shown, such as the input buffer 28, a turbo decoder 40 (a block diagram of which is shown in Figure 5), an error check 42, and an output buffer 30. As those skilled in the art will appreciate, the turbo decoder 40 and error check 42 of Figure 3 each have a separate processor (not shown). The frame is decoded pursuant to aspects of the invention as described below, and the decoded frame is then forwarded.

[0042] Alternate embodiments of the pertinent portions of the receiver may be structured as shown in Figure 4. Figure 4 shows a processor 46, which can be for example a microprocessor, which functions as the controller for the error check 26 and the decoder 24 and storage devices 28, 30.

[0043] The performance of iterative and turbo decoders improve as the number of iterations increase, however, the computational complexity also increases almost linearly with the number of iterations. This can cause problems in decoders that have slow decoding or processing capabilities.

[0044] The number of iterations that the turbo decoder requires to converge varies from frame to frame for a typical channel. Once convergence is reached, the hard decision output of the turbo decoder will not change. If the maximum number of iterations for convergence is fixed (*N*) per frame, the convergence for certain frames that don't require N iterations for convergence would be reached within a fewer number of iterations (*F*). Therefore, *(N-F)* redundant iterations would have to be performed. For other frames, F iterations might be insufficient for convergence to be reached.

[0045] Pursuant to the first aspect of the inventions, the problems discussed above can be solved by varying the number of iterations per frame. In order to determine when convergence for the frame is reached, an error detection code can be implemented along with the

turbo decoder. It is contemplated that any error detection check that works with a bit stream, or a linear block code that has error detection capabilities may be used. Figure 5 shows a block diagram having two decoder blocks 48, 50. The first decoder block 48 is shown receiving frames from the demodulator 52 of the receiver (not shown), and an error detection check 54 is shown. One example of an acceptable error detection check 54 is a Cyclic Redundancy Check (CRC) check. If the error detection check 54 does not detect errors, the frame is forwarded, otherwise the frame is sent back to the decoder blocks 48, 50 for further decoding.

[0046] Pursuant to the first aspect of the invention, the error detection check 26, 42 can be used after each decoding iteration has been performed. Alternatively, the error detection check 26, 42 may be used after a predetermined number of iterations has been performed. The error detection check 26, 42 is used to determine if there are any errors in the frame being decoded. If no errors are detected, then iterative decoding for that frame can be stopped, thus saving time and processing power by preventing unnecessary decoding iterations.

[0047] For example, figure 6 shows the plot of a BER (Bit Error Rate) and a FER (Frame Error Rate) for a frame length of 1280 bits after performing 12 iterations for each frame. The same performance, in terms of BER and FER, can be achieved with a lower average number of iterations. Figure 7 shows the average number of iterations required to achieve the same performance. Thus, if an error detection check 26, 42, such as, for example, a CRC, is used, after each iteration or after a predetermined number of iterations, the complexity and power consumption of the decoder can be reduced. Consequently, the decoding speed may be increased by performing only the minimum number of decoding iterations required for substantial convergence on each frame as opposed to performing unnecessary iterations for frames that don't require the full 12 iterations.

[0048] Another problem can arise if the decoder cannot perform enough iterations for substantial convergence within the system's decoding time constraint per frame, which can be, for example, a frame duration (10 ms). The simulation results of the number of iterations required for frames of length 1280 bits for Eb/No values of 0.5, 0.75, and 1.0 dB are shown in Figures 8-10, respectively. It is apparent that the number of iterations required for certain frames is greater than the average number of iterations (the number of iterations required for the average

frame), and that the number of iterations required for other frames is less than the number required for the average frame.

[0049] For systems having decoding time constraints per frame, it may not be possible for the decoder to decode a frame that requires a greater number of iterations than the average frame within the time constraint, while other frames, which require less than the average number of iterations, could be decoded well within the time constraint. This can particularly apply to 3G systems because of the time constraints they are subject to.

[0050] If a system is subject to a time constraint for decoding a frame, and thus limited to a maximum number of iterations it can perform for each frame based on the time constraint and the decoder's processing speed, the BER and FER performance of the decoder may degrade. For example, if a decoder is subject to a one frame time constraint of, for example 10 ms, and the decoder is only capable of computing 4 iterations every 10 ms, the maximum number of iterations is also fixed at 4. This can cause significant degradation in FER and BER performance. To illustrate, because of the 10 ms time constraint and the decoder's 4 iterations/10 ms processing speed, the decoder could only perform 4 iterations for corrupted frames that may require, for example, 6 iterations for substantial convergence. If only 4 iterations are performed for a corrupted frame that requires 6 iterations, a greater FER and BER could result.

[0051] Figure 11 shows FER as a function of the number of iterations, and shows that, for Eb/No value of 0.75 dB, the FER after 4 iterations is around 0.1 whereas it is about 0.002 after 12 iterations.

[0052] Pursuant to the first aspect of the invention, frames received from the channel can be stored or buffered, resulting in improved FER and BER performance. To illustrate, if a decoder is capable of computing 4 iterations per frame duration, from Figure 9, one can see that the average number of iterations for an Eb/No value of 0.75 dB is around 4 and the FER without storing or buffering is about 0.1 (from figure 11). If input storage devices or input buffers are used, one can see that an FER of about 0.002 for the same Eb/No value can be achieved while maintaining the average decoding time well within the frame duration time constraint. As discussed below, output storage devices or output buffers 30 could also be used to facilitate the specific requirements of upper layers that subsequently process the decoded frames.

[0053] The increase in performance occurs because only the necessary number of iterations are performed for each frame to reach substantial convergence, and a storage device or buffer 28 is used to store other frames when the decoder 24, 40 is busy decoding a frame that requires "extra time" to decode ("extra time" refers to any time period that is greater than the system's decoding time constraint per frame). For example, if the maximum number of iterations is increased and a buffer 28 is incorporated, a subsequent frame may be stored in the buffer while the decoder 24, 40 spends extra time to decode a corrupted frame. After the requisite number of iterations is performed on the corrupted frame, the decoder 24, 40 may proceed to decode the next frame, which may be uncorrupted or average.

[0054] For systems having a decoding time constraint per frame of, for example, 10 ms, if the maximum number of iterations was increased and fixed beyond the number of iterations the decoder could perform in 10 ms, each subsequent frame would be dropped if a storage device or buffer was not used. If the maximum number of iterations was fixed and the number of iterations was variable, on the other hand, subsequent frames would be dropped when the decoder performed extra iterations for corrupted frames that required more than 10 ms to decode.

[0055] Pursuant to methods and embodiments incorporating the first aspect of the invention, as shown in Figures 3 and 4, an error check 26, 42 and an input storage device or input buffer 28 can be used with an iterative decoder 24, 40 to produce beneficial results. In the first embodiment and method, a corrupted frame that requires, for example, 6 iterations, may be decoded in 15 ms by a decoder 24, 40 that has a processing speed of 4 iterations/10 ms. If the system was subject to a decoding time constraint of, for example 10 ms, a subsequent frame, which could be uncorrupted and require only 2 iterations, could be stored or buffered in the storage device or buffer 28 while the decoder 24, 40 is busy decoding the corrupted frame. The decoder 24, 40 could then retrieve and process the uncorrupted frame from the storage device or buffer 28 in 5 ms after decoding the corrupted frame. Consequently, the average decoding time per frame would meet the 10 ms decoding time constraint per frame while achieving optimal BER and FER performance. In the embodiments shown in Figures 3 and 4, the error or convergence check 26, 42 allows for a variable iterative decoder 24, 40 while an input storage device or buffer 28 prevents subsequent frames from being dropped while the decoder is busy

decoding corrupted frames that require more than 10 ms (i.e., "extra time") for decoding. In other embodiments not shown, a storage device or buffer may be used with or without an error check.

[0056] It should be noted that, for the embodiment shown in Figure 3, the processor (not shown) of the error check 42 performs the error checks and indicates when a frame is to be sent from the error check 42 to the decoder 40. The processor (not shown) of the decoder 40 performs the decoding iterations and indicates when to retrieve frames from the input storage device or input buffer 28 and begin decoding them, when to send frames that have had decoding iterations performed on them to either the error check 42 or the output buffer 30, and also indicates when the output buffer is to output frames to an upper layer (not shown).

[0057] It should also be noted that, for the embodiment shown in Figure 4, the processor 46 is what commands the transfer of frames from one block to another (for example, from the input buffer 28 to the decoder 24), and performs the decoding operations and convergence checks.

[0058] A first variation of methods and embodiments incorporating the first aspect of the invention can include an upper limit for the number of iterations. From inspecting Figure 11, one can see that the average number of frames that can be corrected beyond a certain number of iterations decreases considerably as the number of iterations increases. Consequently, pursuant to the first variation, the upper limit should be fixed based on the timing of the turbo decoder as well as the overall performance of the system. For all the simulation results given, the upper limit was 12. The input and output storage devices or buffers of the turbo decoders can be implemented as a FIFO queue. For a certain maximum number of permitted iterations, as well as for a particular Eb/No value and framelength, a statistic of the average number of iterations taken by frames can be determined.

[0059] If the decoder can compute the average number of iterations within the system's decoding time constraint per frame (for example, 10ms), the average service rate will be equal to or greater than the arrival rate of frames, and thus the input queue would reach a steady state and follow Little's theorem. In this case, the average output rate from the decoder will be equal to the average arrival rate in the input queue.

[0060] If the upper layer reads frames from an output storage device or buffer at the same rate as arrival, the output queue will also reach a steady state. If the decoder speed is increased, the occupancy of input and output storage devices or buffers will change. Figure 12 shows a plot of the input storage device or buffer occupancy given for a turbo decoder capable of performing 4-4.5 iterations for a frame length of 1280 bits within 10ms. The Eb/No value is 0.75db, and the average number of iterations required is 3.9. From this figure the size of the required buffer can be estimated. Note that pursuant to other aspects of the invention, output storage devices or output buffers are used in conjunction with the decoder to facilitate specific requirements that upper layers might have.

[0061] In sum, pursuant to the first aspect of the invention, the average time to decode all frames is less than or equal to the decoding time constraint per frame. This is done by increasing the number of decoding iterations for frames that require a greater number of decoding iterations for substantial convergence. The error check determines whether there are errors in the frames, and if there are, additional decoding iterations should be performed on those frames. In the meanwhile, while additional iterations are performed on those frames requiring a greater number of decoding iterations, other frames are stored or buffered for subsequent decoding.

[0062] Pursuant to a second aspect of the invention, the buffer or storage device size limit is determined so that the probability of storage device or buffer overflow is within an acceptable limit. Storage device or buffer overflow occurs when the number of frames stored or buffered in the storage device or buffer exceed the buffer size limit, in which case frames that cause the buffer to exceed the buffer size limit are dropped.

[0063] One example of a way to determine an acceptable storage device or buffer size limit is described below. The storage device or buffer size limit may be premised on various parameters and constraints, such as, for example, storage device or buffer overflow or memory constraints.

[0064] In this example, the storage device or buffer size limit is based on the probability of storage device or buffer overflow. For example, let F_b be the FER due to buffer overflow and F_e be the FER due to error and $F_t = F_b + F_e$, where F_e is known. Let $F_b <$ F_e/(precision factor) and L be the buffer length to be fixed. Then find L to satisfy the equation listed below.

$$Fb = 1-\Sigma_{i=0 \text{ to } L}Pr(b=i)$$

[0065] Pursuant to the example, Figure 13 shows the maximum storage device or buffer length required for the same Fb at different processing speeds. Figure 14 depicts a plot for the input waiting time profile. From this graph, one can see that for a decoder speed of 4 iterations per 10ms, the waiting time probability for a 13 frame duration is 5.0e-06. If the maximum input storage device or buffer size is maintained as the profile shown in figure 13, the storage device or buffer drop profile will be the profile shown in figure 15.

[0066] Pursuant to a third aspect of the invention, frames are decoded and processed out of sequence depending on the number of iterations required to decode the frames. This desirably prevents frames that require a greater number of decoding iterations to reach substantial convergence from stopping other frames that require less decoding iterations from being decoded. A third configuration of receiver components pursuant to the third aspect of the invention is shown in Figure 16. Note that although an input storage device or buffer 28 is shown in Figure 16, embodiments incorporating the third aspect of the invention need not include an input storage device or buffer.

[0067] Frames are decoded in the decoder 24 and are checked for errors by the error check 26 after a predetermined number of decoding iterations have been performed by the decoder. If an error is detected, the frames having errors are sent to the alternate storage device or alternate buffer 62 from the error check 26 for supplemental decoding at a subsequent time. During that time, frames that are received from a channel or frames previously received from the channel and stored or buffered in the input storage device or buffer 28 are decoded by the decoder 24. The subsequent time is typically when the decoder 24 is not receiving other frames from the channel or when the input storage device or buffer 28 has a predetermined amount of memory available. Frames stored or buffered in the alternate storage device or alternate buffer 62 are then sent to the decoder 24 for supplemental decoding out of sequence. Note that supplemental decoding refers to performing more decoding iterations after the predetermined number of iterations have been performed. The error check 26 rechecks, either iteratively after each supplemental decoding iteration or after a predetermined number of supplemental decoding iterations, for errors. If no error is detected, the frames are resequenced by a sequencer 64 and stored or buffered in the output storage device or output buffer 30. Note that the sequencer can

resequence the frames in a variety of ways, for example, by using a look-up table that is based on a temporal frame reference.

[0068] It should be noted that, for the embodiment shown in Figure 16, the processor 46 is what commands the transfer of frames from one block to another (for example, from the alternate buffer 62 to the decoder 24), and performs the decoding operations and error checks.

[0069] Pursuant to a fourth aspect of the invention, the decoded frames can be retrieved by an upper layer at a constant rate, or pursuant to any process or scheme whose average rate is equal to or less than the rate frames are decoded, even though the decoder decodes the frames at a variable rate. Decoded frames that, for example, are substantially converged, can be stored in the output buffer 30 and later sent to upper layers, as shown in Figures 3, 4, and 15. This allows for the output storage device or output buffer 30 to send the decoded frames to the upper layer at a constant rate, or pursuant to any process or scheme whose average rate is equal to or less than the rate frames are decoded, even though the frames are decoded at a variable rate. Note that an upper layer can be any information recovery process.

[0070] Note that, throughout the disclosure, the terms "stored" and "buffered" have been used in the alternative because the action to be performed can be either to "store" or to "buffer." Likewise, throughout the disclosure, the terms "storage device" and "buffer" have been used in the alternative because the component described can be either a "storage device" or a "buffer," based upon the frame tracking capabilities of an associated controller.

[0071] From the foregoing it will be observed that numerous modifications and variations can be effectuated without departing from the true spirit and scope of the novel concepts of the invention. It is to be understood that no limitation with respect to the embodiments illustrated is intended or should be inferred.